

Scalable Data Collection Infrastructure for Digital Government Applications *

Hanan Samet Egemen Tanin

Department of Computer Science
Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland at College Park

hjs@cs.umd.edu egemen@cs.umd.edu

Leana Golubchik

CS and EE-S Departments
Integrated Media Systems Center
Information Sciences Institute
University of Southern California

leana@cs.usc.edu

Data and interactions with data are becoming less dependent on centralized systems. In recent years, users have gained access to large volumes of data over the Internet. Transferring large volumes of data and working with it is a challenge. Many servers that make large volumes of data available over the Internet regularly slow to a crawl with a few bulk data transfer requests. Within the context of this project we focussed our efforts in two directions.

First, we built BISTRO as a recent framework for scalable and secure wide-area upload applications. For example, deadline driven online tax form submissions forms a prime application for this framework. BISTRO uses intermediaries (termed Bistros) for improving the efficiency and scalability of uploads while maintaining privacy and integrity of the collected data. BISTRO's approach is to break the original deadline-driven upload problem into the following steps: (i) a real-time timestamp step, (ii) a low latency commit step, where the data goes to an intermediary, and (iii) a timely data collection step, which can be carefully planned (and coordinated with other uploads) and results in efficient data delivery to the original destination while maintaining the necessary security characteristics.

Second, we have been developing a sample interactive data browser for access to large online databases. Specifically, our application domain contains large volumes of online spatial data that require high levels of visual interactions. We define different types of usages for our browser. The browser can be activated as an applet so that users across various platforms can interactively access a database at a remote location. Also, the browser along with an Internet-enabled database management system can be installed on the local architecture for more prolonged usages of the browser. In this case, the browser can still be utilized to view data from remote locations. However, in this second type of usage, the data that will be frequently used can be downloaded to the local database on demand, and subsequently accessed locally with high levels of continuous interactivity.

Within the context of the current project, we are working on two subdirections for downloads that we plan to incorporate the results of to our interactive browser. First, we want to help users that wish to manipulate large volumes of online data by developing APPOINT (an Approach for Peer-to-Peer Offloading the INternet). APPOINT is a new peer-to-peer approach to provide users with the ability to transfer large volumes of data more efficiently by better utilizing the distributed network resources among active clients of an existing client-server architecture. We developed a library of functions, with a simple application programming interface (API), that can be easily plugged into an existing system. With APPOINT, a server still exists and it is the central source for the data and the decision center for the service. The environment still functions as a client-server environment under many circumstances. Yet, APPOINT continuously maintains

*This work was supported in part by the US National Science Foundation under Grant EIA-00-91474.

various types of information on the clients in the background. This includes, information on what each client has been downloading, their availabilities, bandwidths, etc. When the client-server service starts to perform poorly or a request for a data item comes from a client with a poor connection to the server, APPOINT can start appointing appropriate active clients of the system to serve on behalf of the server, i.e., clients who have already volunteered their services, have that data from a previous download, and can take on the role of peers (hence, moving from a client-server scheme towards a peer-to-peer scheme). In this scheme, clients are used mainly for the purpose of sharing their networking resources rather than introducing new content and hence they help offload the server and scale up the service. We have built a simulation based evaluation environment for APPOINT. Our goal is to see how much, in terms of average download time, APPOINT can improve a client-server system. We ran multiple experiments under various conditions to observe the reduction in average download time on a client-server system when APPOINT is used. Our simulations show that we can dramatically improve the performance of current client-server based database systems under various conditions. This is true even for smaller sized data files than we have initially anticipated. Also, while achieving our goal, we showed that we do not have extensive expectations from the peers of our network. APPOINT improves service even under low data file inventory levels made public to the community from the active clients. We have also observed the benefits of APPOINT even for a small number of clients that request data from a server.

Also, we realized that online databases can use many other ideas from the peer-to-peer (P2P) applications for scalable exchange of data. For example, distributed database systems that are loosely connected to each other that host and manage large data sets, Geographic Information Systems (GIS) that can work over the Internet to connect to multiple hosts and visualize/manipulate complex data, and other similar applications can use the ideas from the P2P world to create completely decentralized efficient data exchange environments. The bottleneck for current P2P applications is that the keyword-based searches over P2P networks do not provide the necessary functionality to perform many types of queries. First, one cannot search within the data (i.e., a file). Second, one cannot use many of the attributes of the data for the queries (i.e., except the file name). For a P2P application to yield the desired functionality, which is readily available on many centralized systems, it has to have the capability to facilitate queries, for example in a GIS, such as selecting all the available data from a given region on a US Census map. Various versions of such complex queries can be generated without much effort and in combination with other attributes of the data. For P2P networks, recently developed hashing-based distributed indices address the base problem of querying complete distributed data without a central authority. For example, one can provide the file name of a music file that is to be downloaded over a P2P network. Accessing such information is facilitated by the use of an index. These indices, although being quite scalable, do not support more complex queries such as rectangle intersections and, more generally, range queries, which are fundamental to many data management systems. Their mechanisms heavily rely on creating globally known mappings between the node addresses of a network and the data file names that are available in this network. To avoid all-to-all communications, the mapping functions should be known by all the peers of the network. In our work, we introduce a distributed hashing-based index that facilitates responding to complex queries, specifically on spatial data, over P2P networks. Our initial experiments showed that our index can work well under many circumstances. This work can efficiently be used in future Digital Government applications to facilitate complex queries on complex data for decentralized settings. In essence, we realized that a distributed system requires assigning responsibility for regions of space to the peers in the system and making this assignment globally and implicitly known to other peers. Hence, our work builds up on the distributed hashing research but uses regions of space rather than object names/ids.